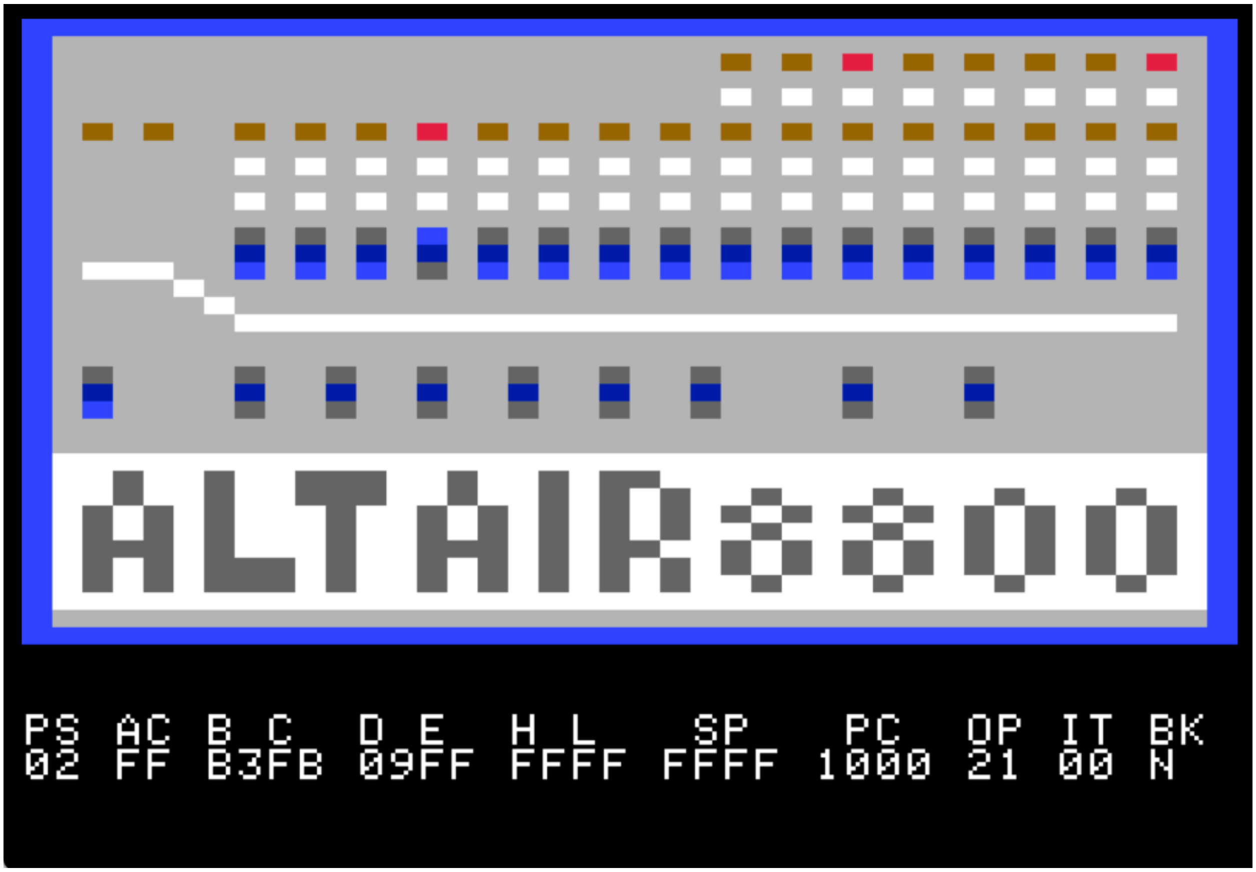
SIM-8800: Altair 8800 emulator for Apple II



```
PS AC B C  D E  H L   SP   PC   OP IT BK
02 FF B3FB 09FF FFFF FFFF 1000 21 00 N
```

Sim-8800 is based on APPLE-80 ©1979 by Dann McCreary.

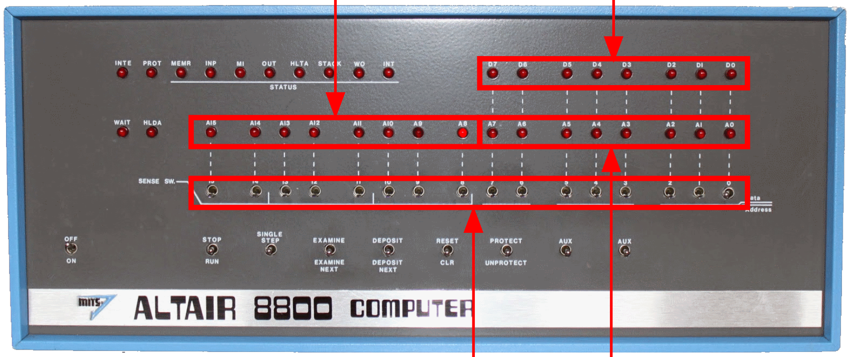
Additional operating instructions can be found in the APPLE-80 manual.

Release 1.0b

SIM-8800 Altair display mode

Current Byte/Opcode
D7-D0

Address High byte
A15-A8

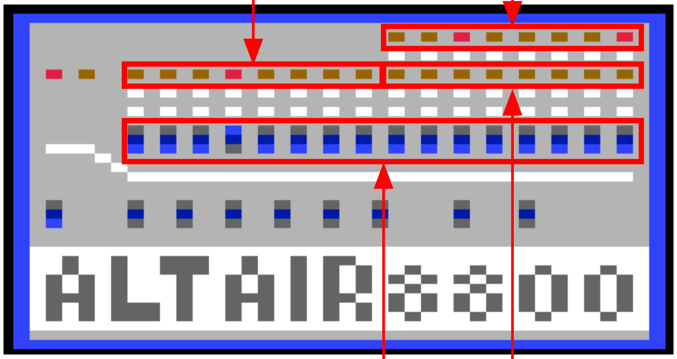


Switches 15-0

Address Low byte
A7-A0

Current Byte/Opcode
D7-D0

Address High byte
A15-A8



Switches 15-0

Address Low byte
A7-A0

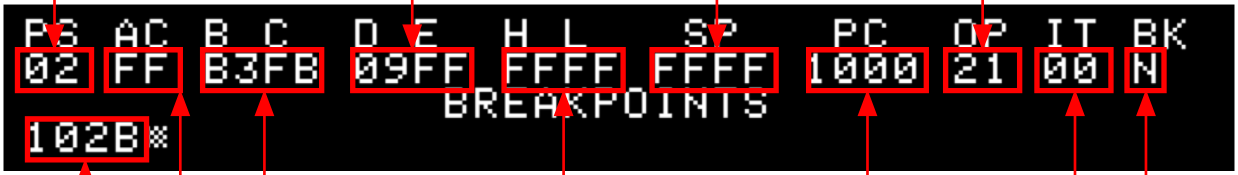
8080 Status line

D/E Register pair

Current 8080 Opcode

Processor Status byte

Stack Pointer



```
PS AC B C   D E   H L   SP   PC   OP IT BK
02 FF B3FB 09FF FFFF FFFF 1000 21 00 N
                BREAKPOINTS
102B※
```

Current breakpoints

H/L Register pair

Breakpoints set Y/N

Accumulator byte

Program Counter

B/C Register pair

Interrupts Enabled

Startup:

At startup, Sim-8800 will display a condensed set of instructions. Press the space bar to continue to the main Altair interface mode. Press **I** any time the virtual Altair is paused to view the instruction screen again.

Keyboard commands:

0-9, A-F TOGGLES SWITCHES 0-15
Number keys 0-9 and letters A-F will toggle the corresponding switch on the Altair, numbered 15-0 from left to right. When the program starts, switch 12 (C) is set up, to indicate the starting memory location of $1000

X TO EXAMINE MEMORY
The byte in memory at the address indicated by the address switches is displayed in the data LEDs. e.g, when the program starts, the address is set to $1000, so pressing X will display the byte at $1000 (21).

N FOR NEXT ADDRESS, **^** FOR PREVIOUS
N will advance the program counter address by one, and display the byte at that address. The caret (^) will decrement the address and display the corresponding byte.

P TO DEPOSIT BYTE INTO MEMORY
To write a byte to memory, set switches 15 through 0 to the corresponding address, and examine that byte. Then set switches 7 through 0 to the byte value you want to write, and press P to deposit it into memory.

O TO DEPOSIT NEXT BYTE INTO MEMORY
Once a byte is set into memory, another can be deposited at the next address in memory with the O key. This increments the program counter and writes the byte value set with switches 7 through 0 into memory.

SPACE OR **S** TO EXECUTE 1 8080 PROGRAM STEP
This will execute the opcode at the currently displayed program counter address, then pause the machine. The results of the operation are displayed in the status line and the LEDs will update with the new address and data bytes.

T TO START INTERACTIVE TRACE MODE
This will execute a program, one opcode at a time, while updating the LED display and the status line. While tracing a program, you can toggle the switches (emulated input 00). The trace will continue until interrupted with control-C or the program reaches a breakpoint.

G TO RUN PROGRAM IN NON-INTERACTIVE MODE
In this mode, the program will run, full speed, until interrupted by control-C or a breakpoint is reached. Switches and LEDs will not be active during this “non-interactive” mode.

CONTROL-C TO STOP RUN/TRACE MODE
While a program is running, it can be interrupted with control-C, and the current program counter and register status will be displayed in the status line and LEDs.

K TO SET BREAKPOINTS
Pressing K while the Altair is paused will display the word “Breakpoints” below the status line. You may then enter or edit several memory addresses to be breakpoints for your program separated by spaces. Type in the addresses, or use the arrow keys to move the cursor over an existing breakpoint. Press enter/return when done. When the program reaches a breakpoint addresses while tracing or running, it will pause the Altair and display the registers and program counter status.

R TO EDIT REGISTERS
While the Altair is paused, the contents of the registers and accumulator can be edited. Use the arrow keys to move the cursor over the register, and type the new value. Press return to accept the new register values.

I FOR INSTRUCTIONS
When the virtual Altair is paused, and not running a program, pressing I will display the instructions screen. Pressing Space will hide the instructions and return to the virtual Altair display.

Kill the Bit

Included in Sim-8800 is the classic Altair 8800 demonstration program, “Kill the Bit.”

As Sim-8800 starts up, it moves the 8080 code to address $1000. If you inspect the program on disk, you will see the 8080 opcode bytes at the end of the Sim-8800 code ($306E).

```
;         classic "kill the bit" relocated to $1000
; 1000                   org 1000
; 1000    21   00   10        lxi h,0       ;initialize counter
; 1003    16   80             mvi d,080h    ;set up initial display bit
; 1005    01   0E   00        lxi b,0eh     ;higher value = faster
; 1008    1A             beg: ldax d   ;display bit pattern on
; 1009    1A                  ldax d        ;...upper 8 address lights
; 100A    1A                  ldax d
; 100B    1A                  ldax d
; 100C    09                  dad b         ;increment display counter
; 100D    D2   08   10        jnc beg
; 1010    DB   00             in 00h        ;input data from sense switches
; 1012    AA                  xra d         ;exclusive or with A
; 1013    0F                  rrc           ;rotate display right one bit
; 1014    57                  mov d,a       ;move data to display reg
; 1015    C3   08   10        jmp beg       ;repeat sequence
; 1018                        end
```

Before tracing the program, set switches 15-8 (F-8) off. As the program traces, you will see one LED in the Address High Byte bank of LEDs light up and move across. The goal is to “kill” the bit by flipping up the switch under the lit LED while it is still lit, then flip it down again once the LED turns off. If you miss, another bit is added to the pattern for you to kill.

Additional Altair 8800 programs can be found online that take advantage of the switches and LEDs as input and output. To get them to run on the virtual Altair, a few considerations must be made:

- Relocate the code to start at $1000 instead of $0000, since the emulator and the Apple II both use pages 00-0F for their own code and variables.
- Sim-8800 switches are input 00, not FF.
- Sim-8800 LEDs are output FF, otherwise they are set to display the byte in register D.